Learning Abstract Skillsets with Empowerment Bandits

Anonymous Author(s)
Affiliation
Address
email

Abstract

General purpose agents will need to be able to execute diverse skillsets in stochastic settings. Empowerment is an appealing objective for helping agents build large skillsets in stochastic environments because empowerment can enable agents to learn abstract skills that target groupings of states. Yet existing empowerment methods have been unable to fulfill this potential for learning large abstract skillsets in stochastic settings. A key reason is that the objective encourages stagnant skillsets rather than increasingly diverse skillsets. To overcome this issue, we introduce a modified empowerment objective that converts the objective into a bandit problem that directly encourages diverse skillset learning. The bandit policy outputs a skillset in the form of the set of parameters that make up the skill-conditioned policy. The reward is the mutual information between skills and states of the skillset action, which measures the diversity of the skillset action. We show empirically that our approach is able to learn large abstract skillsets in stochastic domains, including ones with high-dimensional observations.

1 Introduction

General purpose agents that operate in the real world will need to be able to execute a diverse set of skills in a highly stochastic world. A future self-driving car will need to be able to execute the multitude of skills required in driving (e.g., changing speeds, turning, changing lanes, etc.) while nearby cars, pedestrians, and bicyclists continually make random movements. Similarly, a household robot will need to be able perform its large number of household chores while the human members of the household randomly come and go and randomly converse with each other and the robot. For a household robot, even the simple act of turning its head to look in a different direction can be highly stochastic as the robot will often see objects in unexpected places.

A major problem in AI research is that it unclear whether the dominant paradigms in unsupervised skill learning are capable of learning large skillsets in environments with realistic levels of randomness. The dominant approach to unsupervised skill learning, unsupervised Goal-Conditioned RL (GCRL), which trains agents to learn skills that target particular regions of the spate space, has repeatedly demonstrated that it can learn diverse skillsets in deterministic settings where specific regions of the state space can be consistently achieved (Ecoffet *et al.*, 2019; Mendonca *et al.*, 2021; Nair *et al.*, 2018; Pong *et al.*, 2019; Campos *et al.*, 2020; Pitis *et al.*, 2020; Held *et al.*, 2017; Kim *et al.*, 2023). However, it is unclear how unsupervised GCRL will perform in settings with more realistic levels of randomness, in which specific regions of the state space cannot be regularly achieved. In these settings, there can be little to no reward signal for a skill trying to reach a specific state.

Empowerment (Klyubin *et al.*, 2005; Salge *et al.*, 2013; Jung *et al.*, 2012; Mohamed & Rezende, 2015; Gregor *et al.*, 2016; Eysenbach *et al.*, 2018), another approach to learning large skillsets, does offer some mathematical advantages when it comes to skill learning in stochastic domains. As the

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

maximum mutual information between skills and states, empowerment encourages agents to learn a large set of skills, in which each skill targets a distinct grouping of states. These grouping of states can be small such as in more deterministic settings where specific states can be achieved. Or they can be larger such has in more stochastic settings where only bigger sets of states can be targeted (see Figure 1 for an illustration). Thus, instead of forcing agents to learn skills that target specific states like GCRL, empowerment enables agents to learn more abstract skills that can target sets of states that are not necessarily close to one another in the state space. For instance, empowerment can help a household robot learn a skill to open a door. The skill could target a grouping of states, in which all states in the group describe a scenario in which the robot has opened the door and is looking inside the next room. At the same time, each state in the group can differ in terms of the number of people, set of objects, and lighting that the robot sees in the next room.

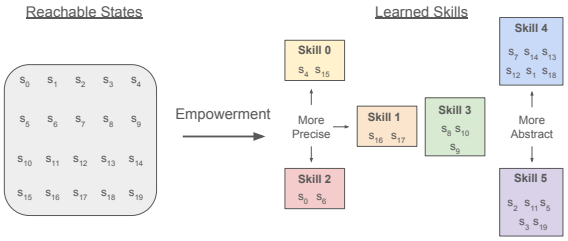

Figure 1: Illustration of how empowerment enables agents to learn distinct skills by encouraging each skill to target a unique grouping of states. In relatively deterministic settings, skills will target smaller, more precise groupings of states, while in more stochastic settings, skills will target larger, more abstract groupings of states.

Yet empowerment empirically has not been able to achieve its potential for learning large abstract skillsets in stochastic settings. Even in deterministic settings, several prior works have noted that empowerment tends to produce stagnant policies that do not expand beyond the skills the agent has at initialization (Levy *et al.*, 2023; Campos *et al.*, 2020; Anonymous *et al.*, 2024). The key problem with the typical implementation of the empowerment objective is that it does not encourage diverse skillset learning (Levy *et al.*, 2023; Campos *et al.*, 2020; Anonymous *et al.*, 2024). Empowerment is typically optimized with Goal-Conditioned RL (GCRL) in which the reward for reaching a state while pursuing a certain goal is the probability that the goal that is being pursued is the agent's goal given the state that is reached and given the current skill-conditioned policy. In other words, given a goal, the agent is rewarded for going to states that goal typically goes to so there is little incentive to change the skill-conditioned policy to try to add new skills.

We introduce a new empowerment objective that explicitly encourages diverse skillset learning. The objective takes the form of a bandit problem, in which the bandit policy outputs the set of parameters that make up the skill-conditioned policy neural network (i.e., the policy outputs a skillset). The reward for the proposed skillset is the mutual information between skills and states produced by the proposed skillset (i.e., the reward measures the diversity of the skillset action). We validate our empowerment objective in several highly stochastic, but otherwise simple domains, including ones with high dimensional observations. While our approach is able to learn large skillsets, both GCRL and a popular empowerment method were not able to learn a skillset with any meaningful size. To our knowledge, our approach is the first unsupervised skill learning to learn skillsets in stochastic settings.

Our approach does have a critical limitation in that it assumes the agent has access to a simulator of the environment as the algorithm requires a prohibitive amount of interaction with the environment. However, recent work by Anonymous *et al.* (2024) (paper also submitted to RLBRew) introduced some modifications to our Empowerment Bandits objective enabling the agent to reuse its past transition data, which makes the amount of environment interaction tractable. Indeed, Anonymous *et al.* (2024) were able to match our results despite not requiring a simulator.

2 Background

2.1 Problem Setting

We assume the typical unsupervised skill-learning problem setting in which an agent interacts in a controlled Markov process, which is a Markov Decision Process (MDP) without a reward function. The controlled Markov process is defined by the tuple $(\mathcal{S}, p(s_0), \mathcal{A}, p(s_{t+1}|s_t, a_t))$, in which $\mathcal{S}$

is the space of states; $p(s_0) : \Delta(\mathcal{S})$ is the initial state distribution; $\mathcal{A}$ is the space of actions; $p(s_{t+1}|s_t, a_t) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ represents the transition dynamics distribution.

The goal of the unsupervised skill-learning setting is for agents to learn a large skillset that can be used for downstream tasks. We will define a skillset by the tuple $(\mathcal{Z}, l, \theta, p(s_n|s_0, l, \theta, z))$. $\mathcal{Z}$ is the space of skills. l represents the parameter(s) that define the distribution over skills $p(z|l)$. For instance, in our approach the distribution over skills takes the form of a uniform distribution over a d-dimensional cube, and l is the scalar value that reflects the side length of each of the d-dimensions of the cube. In our 2-dimensional tasks, skills are sampled from a square with side length l. θ represents the set of parameters that define the skill-conditioned policy $\pi(a|s, z, \theta)$. In our case, θ is the set of weights and biases that make up the skill-conditioned policy neural network, which takes as input the state and skill and outputs the mean of the distribution over actions. $p(s_n|s_0, l, \theta, z)$ is the transition function that outputs the skill-terminating state s_n given the start state s_0, skill distribution parameter l, skill-conditioned policy parameters θ, and the specific skill z.

2.2 Empowerment

Empowerment is a objective function for learning diverse skillsets. The empowerment of a state s_0 is typically defined

$$\mathcal{E}(s_0) = \max_{l,\theta} I(Z; S_n|s_0) \tag{1}$$

$$= \max_{l,\theta} H(Z|s_0) - H(Z|s_0, S_n) \tag{2}$$

$$= \max_{l,\theta} \mathbb{E}_{z\sim p(z|s_0), s_n \sim p(s_n|s_0,z)}[\log p(z|s_0, s_n) - \log p(z|s_0)]. \tag{3}$$

$I(Z; S_n|s_0)$ is the mutual information between skills and skill-terminating states for the state s_0 under consideration. Note that in line 3, l and θ have been marginalized in the distributions $p(z|s_0) = p(z|s_0, l)$ and $p(s_n|s_0, z) = p(s_n|s_0, z, \theta)$. Thus, the goal in empowerment is to learn a skillset defined by (l, θ) with high mutual information between skills and states. Per line 2, a skillset has high mutual information if it is diverse because diverse skillsets have (i) relatively high $H(Z|s_0)$ (i.e., the number of skills in the skillset is large) and (ii) relatively low $H(Z|s_0, S_n)$ (i.e., skills target distinct states).

2.3 Variational Bounds on Mutual Information

One challenge with empowerment is that the mutual information term within the empowerment objective is a function of a posterior probability $p(z|s_0, s_n)$ that is intractable to compute in domains with continuous state, action, and skill spaces. Mohamed & Rezende (2015) proposed a solution to this problem — use a variational posterior $q_\phi(z|s_0, s_n)$ parameterized by ϕ instead of the problematic posterior $p(z|s_0, s_n)$, and the resulting term forms a variational lower bound on mutual information I^V.

$$I(Z; S_n|s_0) \geq I^V(Z; S_n|s_0,) = H(Z|s_0) + \mathbb{E}_{z\sim p(z|s_0), s_n \sim p(s_n|s_0,z)}[\log q_\phi(z|s_0, s_n)]. \tag{4}$$

The tightness of this bound depends on the KL divergence between the variational and true posteriors (Barber & Agakov, 2003). For instance, if the variational posterior takes its typical form as a diagonal gaussian (i.e., $q_\phi(z|s_0, s_n) = \mathcal{N}(z; [\mu, \sigma] = f_\phi(s_0, s_n))$), and the true posterior also takes a similar form to a diagonal gaussian (e.g., a diverse skillset in which skills target distinct grouping of states), then the bound has the potential to be tight. On the other hand, the bound can be loose for less diverse skillsets in which distant skills z target the similar states s_n.

Another relevant variational bound on mutual information is GCRL. If the mean and variance of the variational posterior are fixed such that the mean is set to s_n and the variance is fixed to some hyperparameter σ_0 (i.e., $q(z|s_0, s_n) = \mathcal{N}(z; [\mu = s_n, \sigma = \sigma_0])$), then the variational lower bound on mutual information would be a GCRL problem in which the reward is 0 for the first $n - 1$ actions and then $\log \mathcal{N}(z; [\mu = s_n, \sigma = \sigma_0])$. Like a GCRL problem, this reward is higher the closer the skill-terminating state s_n is to the goal state z. The interpretation of GCRL as a variational lower bound on mutual information provides some additional evidence that GCRL should struggle to learn large skillsets in stochastic settings. In stochastic settings, it will generally not be the case that skill end states s_n are the same as the original goal state z. Thus, the true posterior $p(z|s_0, s_n)$ will tend to be more entropic in stochastic domains as a larger variety of goal states could have brought the

agent to s_n. On the other hand, the GCRL variational posterior forms a fixed and narrow distribution around the mean s_n meaning that the GCRL posterior places its highest probability that the goal state z that produced s_n must be near s_n. These potentially large difference between the true posterior and the GCRL variational posterior means the KL divergence between the posteriors can be large, which means the GCRL mutual information variational lower bound can be a loose one. The consequence of this loose bound is that GCRL may not be effective at learning skillsets with high mutual information in stochastic domains because with a loose bound, increasing the GCRL mutual information lower bound does not necessarily mean that the true mutual information $I(Z; S_n|s_0)$ of the learned skillset is also increasing. On the other hand, because the typical variational bound learns the mean and variance of the posterior distribution, the typical variational lower bound to mutual information can still form a tight bound to the true mutual information term in stochastic settings.

2.4 Misspecified Objective

In order to find a skillset that produces a large variational mutual information I^V, the typical approach has been to optimize equation 4 with Reinforcement Learning (Gregor *et al.*, 2016; Eysenbach *et al.*, 2018; Achiam *et al.*, 2018; Baumli *et al.*, 2020; Choi *et al.*, 2021). Assuming the distribution over skills $p(z|l)$ is fixed as most of these algorithms do, the objective looks like a goal-conditioned RL problem in which the reward is 0 for the first $n-1$ actions and then the final step reward $R(s_{n-1}, a_{n-1}, s_{n-1}, z, \theta) = \log q_{\phi^*}(z|s_0, s_n)$. ϕ^* is used in place of ϕ because prior to applying RL, an inner optimization loop is executed to update the parameters ϕ so that the variational distribution $q_\phi(z|s_0, s_n)$ is closer to the true posterior $p(z|s_0, s_n)$ and thus the mutual information bound is tighter. ϕ is updated using a maximum likelihood objective $\mathbb{E}_{z\sim p(z), s_n\sim p(s_n|s_0,s_n)}[\log q_\phi(z|s_0, s_n))$. Because the output of this maximum likelihood problem, ϕ^*, is a function of the skill-conditioned policy parameters θ, which are needed to produce the s_n samples in the maximum likelihood problem, that means the reward in the RL problem is also a function of θ, which is why θ was also included as an input into the reward function.

The key problem with applying RL in this fashion is that the objective is not aligned with the goal of trying to learn a set of skill-conditioned policy parameters θ that produces high mutual information. Instead, as noted by others as noted by others (Levy *et al.*, 2023; Campos *et al.*, 2020), the objective is encouraging updates to θ that cause the agent to go states where the agent already visits. This is because the variational posterior $q_{\phi^*}(z|s_0, s_n)$ reflects the relationship between skills and states for the current skill-conditioned policy parameters θ. When $q_{\phi^*}(z|s_0, s_n)$ is then used as a reward and some skill z is being executed, actions that cause the agent to reach states that z frequently goes to will receive more reward. But this can be achieved by making minimal changes to θ. On the other hand, the reward for a new θ that caused many skills to reach new states (i.e., a skillset with higher mutual information), would be $\log 0$, which is undefined.

3 Empowerment Bandits

We introduce a new empowerment objective, which we refer to as Empowerment Bandits (EB), that explicitly encourages diverse skillset learning. Empowerment Bandits makes a couple key changes to the empowerment objective. The first change is to move the l and θ variables from within the distributions $p(z|s_0)$ and $p(s_n|s_0, z)$, respectively, to being conditioned variables in the mutual information term. That is, we will seek to maximize the mutual information term $I^V(Z; S_n|s_0, l, \theta)$ as opposed to $I^V(Z; S_n|s_0)$. If θ is not listed as a conditioning variable, then I^V and q_ϕ are only being trained for the current skillset θ, which will discourage the agent from learning skills that are not contained in the current skillset. If θ is listed as a conditioning variable, then I^V and q_ϕ can be trained for different skillsets, and skillsets with larger I^V can be selected. We also moved the l variable to the set of conditioned variables to give the agent more flexibility to change the size of its skill distribution. In our implementation, l represents the side length of the d-dimensional cube that is the uniform distribution of skills. The second change in Empowerment Bandits is to train policies to output θ and l. That is, EB will learn two policies. One policy f_λ will take as input the skill start state s_0 and skill distribution parameter l and output the parameters of the skill-conditioned policy neural network θ. The other policy f_ψ will take as input the skill start state s_0 and output the skill

distribution parameter l. Thus, the Empowerment Bandits objective can be defined

$$\mathcal{E}(s_0) = \max_{l,\theta} I^V(Z; S_n|s_0, l, \theta) = \max_{l,\theta} H(Z|l) + \mathbb{E}_{z\sim p(z|l), s_n\sim p(s_n|s_0,\theta,z)}[\log q_{\phi^*}(z|s_0, l, \theta, s_n)],$$
$$\theta = f_\lambda(s_0, l),\ l = f_\psi(s_0) \tag{5}$$

Empowerment Bandits optimizes the objective in line 5 as two nested bandit problems. In the inner bandit problem, the agent learns a policy $f_\lambda(s_0, l)$ that outputs a θ and the reward $R(s_0, l, \theta)$ for the skillset is $I^V(Z; S_n|s_0, l, \theta)$. In the outer bandit problem, the agent learns a policy f_ψ that outputs l and the reward is $R(s_0, l) = I^V(Z; S_n|s_0, l, \theta = f_\lambda(s_0, l))$. Figure 4 in the appendix provides a visualization of the two bandit problems.

3.1 Handing the Large Action Space of θ

Optimizing the inner bandit using a typical actor-critic structure, in which the actor is f_λ that outputs a θ action and Q_κ is a critic that approximates $R(s_0, l, \theta)$, will not be feasible due to the large size of θ which can have thousands of dimensions. Both the variational posterior $q_\phi(z|s_0, l, \theta, s_n)$ and the critic $Q_\kappa(s_0, l, \theta)$, which take θ as input, would need to be able to discern the difference in their output from small changes in θ across thousands of dimensions. This would likely be too challenging to learn and/or would require a prohibitive number of skillsets (l, θ) to be tested.

To overcome this issue, insight can be drawn from looking at the gradient with respect to the parameters λ of f_λ assuming a typical actor-critic structure was used. The gradient with respect to one of the parameters in λ would be the dot product of the two vectors (i) $[\nabla_{\theta_0} Q_\kappa(s_0, l, \theta_0), \nabla_{\theta_1} Q_\kappa(s_0, l, \theta_1), \ldots, \nabla_{\theta_{|\theta|-1}} Q_\kappa(s_0, l, \theta_{|\theta|-1})]$ and (ii) $[\nabla_p \theta_0(s_0, l, p), \nabla_p \theta_1(s_0, l, p), \ldots, \nabla_p \theta_{|\theta|-1}(s_0, l, p)]$. The first vector is the gradients of the critic Q_κ with respect to each of the $|\theta|$ parameters in θ. That is, the i-th element of this vector specifies how a small change in θ_i affects Q_κ while the remainder of the parameters in θ remain the same. The second vector is how each θ dimension reacts to a change in p, which is one parameter in λ.

This analysis is helpful because it illustrates that what is needed from a critic $Q_\kappa(s_0, l, \theta)$ is how the reward changes from small changes to a single parameter in θ while the remainder of the parameters remain constant. Thus, one solution to the problem is to learn parameter-specific critics Q_{κ_i} and variational posteriors q_{ϕ_i} for $i = 0, \ldots, |\theta| - 1$. $Q_{\kappa_i}(s_0, l, \theta_i)$ and $q_{\phi_i}(z|s_0, l, \theta_i)$ will approximate $I^V(Z; S_n|s_0, l, \theta_i)$ and the posterior $p(z|s_0, l, \theta_i)$, respectively, in which θ_i represents a θ in which all parameters take on their greedy values from $f_\theta(s_0, l)$ except for the i-th parameter that takes on the value θ_i. Instead of needing to take as input a long θ vector as input, these functions will only take as input the scalar value θ_i. The objective function for updating the parameters λ in f_λ given specific s_0 and l would be:

$$J(\lambda) = \sum_{i=0}^{|\theta|-1} Q_{\kappa_i}(s_0, l, \theta_i),\ \theta_i = f_\lambda(s_0, l)[i]. \tag{6}$$

The gradient of this objective would be the same as the dot product of the two vectors discussed earlier so together all the parameter-specific critics can mimic the gradients from a regular critic $Q_\kappa(s_0, l, \theta)$. In Figure 5, we provide a visualization of how the policy f_λ connects to the parameter-specific critics.

Also, note that updating the parameter-specific critics Q_{κ_i} and variational posteriors q_{ϕ_i} can all be done in parallel with the help of multiple accelerators and the parallelization capabilities of modern deep learning frameworks (e.g., JAX). For instance, in our experiments multiple GPUs were used and each GPU was responsible for computing the updates to hundreds of parameter-specific critic and variational posteriors (see section F for detail on our GPU use). For instance, if $|\theta| = 1000$ and 4 GPUs were available, each GPU could compute the needed gradients for 250 parameter-specific critics in parallel. In terms of computation time, our approach should scale well with large values of $|\theta|$ provided a sufficient number of accelerators are available.

The major limitation that the parameter-specific strategy produces is that it can increase by orders of magnitude the amount of interaction with the environment that is needed. Updating the (non parameter-specific) variational posterior $q_\phi(z|s_0, s_n)$ requires executing a large number of skills in the environment in order to obtain the needed on-policy (z, s_n) tuples. But now with parameter-specific variational posteriors, the on-policy (z, s_n) tuples need to be generated for all $|\theta|$ parameters. This

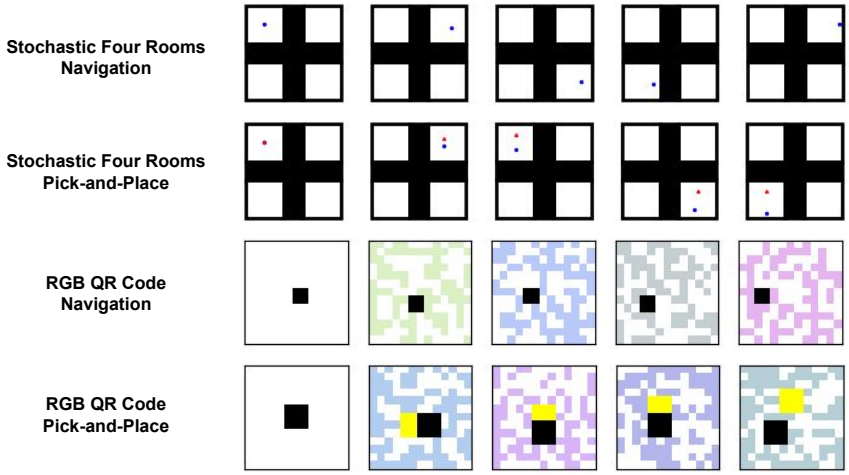

Figure 2: Sample state trajectories executed by a random policy in all four domains.

amount of interaction can be prohibitive, why is why our approach assumes the agent has access to a simulator of the environment. However, recent work by Anonymous *et al.* (2024) (paper also submitted to RLBRew) shows how the mutual information $I^V(Z; S_n|s_0, l, \theta)$ reward can be modified so that prior transition data can be used, which lowers the amount of interaction required by orders of magnitude. Their approach, Off-Policy Empowerment Bandits, builds off of the nested bandit structure that we introduced.

3.2 Algorithm

The Empowerment Bandits algorithm repeats the following three step process. In the first step, the parameter-specific variational posteriors $q_{\phi_i}(z|s_0, \tilde{l}, \tilde{\theta}_i)$ are updated in parallel, in which $\tilde{l}$ and $\tilde{\theta}_i$ are noisy versions of their greedy values. The maximum likelihood objective function for the i-th variational posterior is

$$J(\phi_i) = \mathbb{E}_{\tilde{l},\tilde{\theta}_i,z\sim p(z|\tilde{l}),p(s_n|s_0,\tilde{\theta},z)}[\log q_{\phi_i}(z|s_0, \tilde{l}, \tilde{\theta}_i, s_n)] \tag{7}$$

The purpose of updating the variational posterior is so that the reward I^V can form a tighter bound to the true mutual information.

In the second step, the actor and parameter-specific critics for the inner bandit are updated. The parameter-specific critics are trained via supervised learning to approximate the reward $R(s_0, \tilde{l}, \tilde{\theta}_i)$. The actor f_λ is updated using the objective in line 6. In the third step, the outer bandit actor-critic is updated. The critic Q_ν is updated with supervised learning to approximate the reward $R(s_0, \tilde{l}, \theta = f_\lambda(s_0, \tilde{l}))$. The actor f_ψ is then updated with the objective $J(\psi) = Q_\nu(f_\psi(s_0))$. With this algorithm, the bandit policies are encouraged to output skillsets defined by (l, θ) that are increasingly diverse.

4 Experiments

We perform several experiments to evaluate our two main hypotheses. The first hypothesis we assess is that optimizing our empowerment objective that explicitly encourages diverse skillset learning should learn larger skillsets than prior empowerment approaches. The second hypothesis we evaluate is that it should be difficult for GCRL to learn diverse skillsets in stochastic domains given that the GCRL forms a loose lower bound on the mutual information objective (see section 2.3).

4.1 Environments

Given that existing unsupervised skill-learning approaches like unsupervised GCRL already excel at deterministic tasks, our experiments focused on stochastic tasks and whether agents could still learn

large skillsets when no particular state was achievable with high probability. In addition, because our focus was not on the orthogonal topic of exploration, we used environments in which the abstract states were reachable in a small number of actions. A key constraint on the set of environments we could choose from was that the transition dynamics of the environment needed to be sampled in parallel (i.e., a simulator of the environment needed to be available) due to the large parallel computation requirements of the empowerment algorithms we tested.

To our knowledge, there are no existing benchmarks that satisfy both our stochasticity and JAX-compatibility requirements so we implemented our own domains. We briefly describe the four environments we created next. Additional details are provided in section D of the Appendix. Visualizations of each environment when a random sequence of actions is applied is shown in Figure 2.

1. **Stochastic Four Rooms Navigation:** In this domain, a two-dimensional point agent navigates in an environment with four walled rooms by executing two-dimensional $(\Delta x, \Delta y)$ actions. The domain is highly stochastic because after each action is completed, the agent is moved to the same (x offset, y offset) location in a randomly sampled room. The abstract states that can be targeted are the (x offset, y offset) from the center of a room.
2. **Stochastic Four Room Pick-and-Place:** This is the same environment as the navigation task except there is now an object (the red triangle in the second row of Figure 2) that can be moved if the agent is within a certain distance of the object.
3. **RGB QR Code Navigation:** In this domain an agent learns to navigate amid a continually changing RGB-colored QR code background. Observations are 507-dimensional RGB images and are highly stochastic as the colored-QR code image fully changes after each action.
4. **RGB QR Code Pick-and-Place:** This environment is the same as the navigation task except there is a now an object that can be moved if the object is in reach.

In all domains, there is a single skill start state and skills consist of 5 primitive actions.

4.2 Baselines

We compare our approach, Empowerment Bandits, to both a popular empowerment-based skill-learning method and GCRL. For the prior empowerment-based method, we selected Variational Intrinsic Control (VIC) (Gregor *et al.*, 2016). VIC, like the other popular approaches to optimizing empowerment including DIAYN (Eysenbach *et al.*, 2018) and VALOR (Achiam *et al.*, 2018), use the primitive action space as the learned action space and implement the similar misaligned reward, $\log q_\phi(z|s_0, s_n)$, that encourages the agent to go to states that the current skill-conditioned policy already visits. For the GCRL comparison, our focus was solely on whether goal-conditioned skills can be learned in stochastic domains and not on exploration which is the primary focus on recent unsupervised GCRL algorithms. Thus, we compared against the variant of supervised GCRL that is a lower bound to mutual information discussed earlier. For the higher dimensional QR code tasks, we implemented Reinforcement learning with Imagined Goals (RIG) (Nair *et al.*, 2018). RIG performs GCRL in a latent space learned separately by a VAE. See section E for details on how the baselines were implemented.

4.3 Results

TASK	OURS	VIC	GCRL
FOUR ROOMS NAV.	**5.1**± **0.3**	0.2± 0.4	0.3± 0.4
FOUR ROOM P.-AND-P.	**8.7**± **0.3**	-0.1± 0.3	3.9± 0.6
RGB QR NAV.	**3.5**± **0.1**	-0.4± 0.0	-0.4± 0.3
RGB QR P.-AND-P.	**6.0**± **0.2**	-0.6± 0.1	-2.6± 5.8

Our approach significantly outperforms both baselines in all tasks. Table 4.3, which notes the average variational empowerment (in nats) of the skillsets learned by each baseline, shows that our approach learns a much larger skillset than the baselines in all domains. Note that the baselines sometimes produced negative variational empowerment, which is possible when distant skills target similar skill-terminating states because the variational posterior $q_\phi(z|s_0, s_n)$ needs to learn a distribution over skills more entropic than the true posterior.

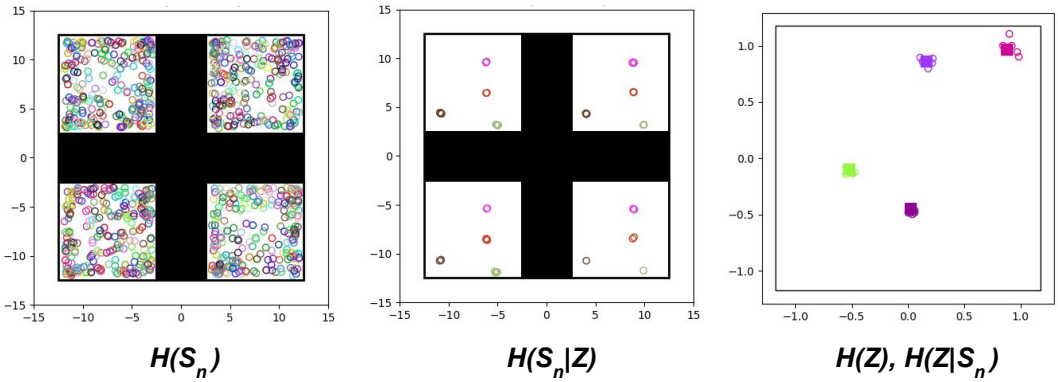

Figure 3: Entropy visualizations for the stochastic four rooms domain. Left image visualizes $H(S_n)$ by marking the skill-terminating state from 1000 skills randomly sampled. The center image visualizes $H(S_n|Z)$ by showing 12 samples of skill-terminating states from 4 specific skills randomly sampled. The right image visualizes (i) $H(Z)$ by showing the skill space (black rectangle) and (ii) $H(Z|S_n)$ by showing samples of the variational posterior (empty circles) for four different skills (filled squares)).

For additional evidence that our approach is able to effectively optimize the empowerment objective, enabling agents to learn large skillsets with the appropriate level of abstraction, we visualize the various entropies that appear in the symmetric definitions of empowerment—$H(S_n), H(S_n|Z), H(Z)$, and $H(Z|S_n)$—for the learned skillsets in all tasks. Figure 3 provides three images showing these visualizations for the stochastic four rooms navigation environment. The left image visualizes $H(S_n)$ by executing 1000 skills uniformly sampled from the learned skill space and marking the skill-terminating state with a colored circle. Per the image, $H(S_n)$ is large as the skill-terminating states produced by the sampled skills nearly uniformly cover the possible state space. The center image visualizes $H(S_n|Z)$ by focusing on four skills, uniformly sampled from the skill space, and for each skill sampling 12 skill-terminating states. Per the image, despite the room the room randomly changing at each transition, each skill targets a particular (x offset, y offset) position (i.e. $H(S_n|Z)$ is low), which is the correct abstract skill in this domain. For example, the brown skill targets a specific position in the bottom left of each room, while the the pink skill targets a position towards the top right of each room. Thus, in this task, the agent is not only learning abstract skills but skills with the appropriate level of abstraction. The image on the right shows samples (empty circles) of the variational posterior distribution, $q_\psi(z|s_0, l, \theta, s_n)$ for four skills (filled squares) sampled from the learned skill space (inner black rectangle). Per the image, $H(Z|S_n)$ is low (i.e., skills are targeting distinct grouping of states) because the samples from the variational posterior form narrow distributions around the sampled skill.

The entropy visualizations for the remainder of the tasks are in section C of the Appendix. For instance, Figure 6 shows the same images for the stochastic four rooms pick-and-place task. The left image shows that $H(S_n)$ is large as the agent is able to execute skills that can achieve many of the possible (agent position, object position) tuples (object is shown by triangles). The center image, which visualizes $H(S_n|Z)$, shows that the agent is learning abstract skills that target (x offset, y offset) positions for both the agent and object. The image on the right visualizes $H(Z|S_n)$ for the now four-dimensional skill space and again the variational posterior forms narrow distributions around the sampled skill showing the agent is learning skills that target distinct groupings of states. Figures 7 and 8 in the Appendix provide the entropy visualization images for the RGB QR code navigation and pick-and-place tasks. Both of these figures show that despite significant stochasticity and high-dimensional observations, our approach learns diverse skillsets with the appropriate level of abstraction.

Moreover, the growth in empowerment from the navigation tasks to the pick-and-place tasks provides further evidence that our approach effectively optimizes empowerment. Moving from navigation tasks to pick-and-place tasks should boost empowerment by a significant factor as for every position the agent can reach, there may be a large number of object positions that can be achieved. The

increase in empowerment by multiple nats confirms that empowerment grew by a large factor when an object was added to the environment.

On the other hand, the baselines were not able to learn large distinct skillsets. For instance, in the stochastic four rooms task, the GCRL agent only learns skills to move to corners of the room as shown in Figure 9, which shows the skill-terminating states of 1000 random skills. More specifically, as shown in Figure 10, when given a goal state of some (x,y) position in one of the four rooms, the agent simply moves towards whichever room the goal is in regardless of where in the room the goal is. This behavior is likely taken to minimize the average distance to the goal as the goal-conditioned reward heavily penalizes the agent if it is far from the specific goal state. In the image-based QR code tasks, the VAE generally struggled to reconstruct the large variety of colored QR codes as shown in Figure 11, ultimately producing an image similar to a mean QR code. The overly abstract latent state space in turn made it challenging for the GCRL component to learn distinct skills. These results provide evidence that GCRL, by forcing skills to target fixed groupings of states, can struggle to learn large skillsets when those fixed groupings are either too specific or too abstract. On the other hand, because empowerment simultaneously learns the skill-conditioned policy and the abstraction (via the variational posterior) in the same objective, empowerment provides the flexibility to adjust the level of abstraction in order to maximize the number of distinct skills learned.

The performance of VIC agents, which optimize empowerment using the primitive action space as the learned action space, also was poor. As with prior works, we also observed stagnant skillsets.

5 Related Work

Our work falls under a large class of algorithms known as Unsupervised RL. Unsupervised RL algorithms seek to gather information about the world (e.g., exploratory data, skills, world models) either without or with limited human supervision in the form of reward functions or manually crafted goal spaces. One large subclass within unsupervised RL are pure exploration methods which seek to cover the state space by maximizing uncertainty in a learned model Pathak *et al.* (2017); Mazzaglia *et al.* (2021); Shyam *et al.* (2018); Sekar *et al.* (2020); Rajeswar *et al.* (2023); Pathak *et al.* (2019); Burda *et al.* (2018) or maximizing state entropy Lee *et al.* (2019); Liu & Abbeel (2021); Yarats *et al.* (2021); Liu & Abbeel (2021). The other major subclass within unsupervised RL is unsupervised skill-learning which includes algorithms that try to learn diverse skill sets without supervision. In addition to the GCRL and empowerment-based skill-learning methods that have been mentioned previously, there are algorithms that combine mutual information objective with exploration bonuses Strouse *et al.* (2021); Park & Levine (2023). There is also another class of algorithms Park *et al.* (2022, 2023a,b) that draw an interesting contrast with empowerment. Instead of taking the empowerment approach of trying to maximize the number of distinct skills learned, these approaches learn small skillsets, but encourage each skill to learn a distinct long-horizon policy.

A key shortcoming of prior unsupervised RL methods is that they have not demonstrated that they can scale to domains with significant stochasticity, in which it is more difficult to learn a model of the transition dynamics or count states or learn skills to achieve particular states. To our knowledge, our approach is the first unsupervised skill learning algorithm to learn skills in stochastic MDPs, in which no state is achievable with high probability.

6 Conclusion

Empowerment has the potential to help general-purpose agents learn large skill sets in stochastic domains, but the objective needs to encourage diverse skillset learning. To this end, we introduce a new empowerment objective that directly trains agents to output diverse skillsets. We show empirically that for the first time an unsupervised skill learning algorithm can learn diverse skillsets in settings with significant randomness.

References

Achiam, Joshua, Edwards, Harrison, Amodei, Dario, & Abbeel, Pieter. 2018. Variational Option Discovery Algorithms. *CoRR*, **abs/1807.10299**.

Anonymous, A., Anonymous, B., & Anonymous, C. 2024. Learning Abstract Skillsets with Empowerment.

Barber, David, & Agakov, Felix. 2003. The IM Algorithm: A Variational Approach to Information Maximization. *Page 201–208 of: Proceedings of the 16th International Conference on Neural Information Processing Systems*. NIPS'03. Cambridge, MA, USA: MIT Press.

Baumli, Kate, Warde-Farley, David, Hansen, Steven, & Mnih, Volodymyr. 2020. Relative Variational Intrinsic Control. *CoRR*, **abs/2012.07827**.

Burda, Yuri, Edwards, Harrison, Storkey, Amos J., & Klimov, Oleg. 2018. Exploration by Random Network Distillation. *CoRR*, **abs/1810.12894**.

Campos, Víctor, Trott, Alexander, Xiong, Caiming, Socher, Richard, Giró-i-Nieto, Xavier, & Torres, Jordi. 2020. Explore, Discover and Learn: Unsupervised Discovery of State-Covering Skills. *CoRR*, **abs/2002.03647**.

Choi, Jongwook, Sharma, Archit, Lee, Honglak, Levine, Sergey, & Gu, Shixiang Shane. 2021. Variational Empowerment as Representation Learning for Goal-Based Reinforcement Learning. *CoRR*, **abs/2106.01404**.

Ecoffet, Adrien, Huizinga, Joost, Lehman, Joel, Stanley, Kenneth O., & Clune, Jeff. 2019. Go-Explore: a New Approach for Hard-Exploration Problems. *CoRR*, **abs/1901.10995**.

Eysenbach, Benjamin, Gupta, Abhishek, Ibarz, Julian, & Levine, Sergey. 2018. Diversity is All You Need: Learning Skills without a Reward Function. *CoRR*, **abs/1802.06070**.

Gregor, Karol, Rezende, Danilo Jimenez, & Wierstra, Daan. 2016. Variational Intrinsic Control. *CoRR*, **abs/1611.07507**.

Held, David, Geng, Xinyang, Florensa, Carlos, & Abbeel, Pieter. 2017. Automatic Goal Generation for Reinforcement Learning Agents. *CoRR*, **abs/1705.06366**.

Jung, Tobias, Polani, Daniel, & Stone, Peter. 2012. Empowerment for Continuous Agent-Environment Systems. *CoRR*, **abs/1201.6583**.

Kim, Seongun, Lee, Kyowoon, & Choi, Jaesik. 2023. *Variational Curriculum Reinforcement Learning for Unsupervised Discovery of Skills*.

Klyubin, A.S., Polani, D., & Nehaniv, C.L. 2005. Empowerment: a universal agent-centric measure of control. *Pages 128–135 Vol.1 of: 2005 IEEE Congress on Evolutionary Computation*, vol. 1.

Lee, Lisa, Eysenbach, Benjamin, Parisotto, Emilio, Xing, Eric P., Levine, Sergey, & Salakhutdinov, Ruslan. 2019. Efficient Exploration via State Marginal Matching. *CoRR*, **abs/1906.05274**.

Levy, Andrew, Rammohan, Sreehari, Allievi, Alessandro, Niekum, Scott, & Konidaris, George. 2023. *Hierarchical Empowerment: Towards Tractable Empowerment-Based Skill Learning*.

Liu, Hao, & Abbeel, Pieter. 2021. Behavior From the Void: Unsupervised Active Pre-Training. *CoRR*, **abs/2103.04551**.

Mazzaglia, Pietro, Çatal, Ozan, Verbelen, Tim, & Dhoedt, Bart. 2021. Self-Supervised Exploration via Latent Bayesian Surprise. *CoRR*, **abs/2104.07495**.

Mendonca, Russell, Rybkin, Oleh, Daniilidis, Kostas, Hafner, Danijar, & Pathak, Deepak. 2021. Discovering and Achieving Goals via World Models. *CoRR*, **abs/2110.09514**.

Mohamed, Shakir, & Rezende, Danilo Jimenez. 2015. *Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning*.

Nair, Ashvin, Pong, Vitchyr, Dalal, Murtaza, Bahl, Shikhar, Lin, Steven, & Levine, Sergey. 2018. Visual Reinforcement Learning with Imagined Goals. *CoRR*, **abs/1807.04742**.

Park, Seohong, & Levine, Sergey. 2023. *Predictable MDP Abstraction for Unsupervised Model-Based RL*.

Park, Seohong, Choi, Jongwook, Kim, Jaekyeom, Lee, Honglak, & Kim, Gunhee. 2022. Lipschitz-constrained Unsupervised Skill Discovery. *CoRR*, **abs/2202.00914**.

Park, Seohong, Lee, Kimin, Lee, Youngwoon, & Abbeel, Pieter. 2023a. *Controllability-Aware Unsupervised Skill Discovery*.

Park, Seohong, Rybkin, Oleh, & Levine, Sergey. 2023b. *METRA: Scalable Unsupervised RL with Metric-Aware Abstraction*.

Pathak, Deepak, Agrawal, Pulkit, Efros, Alexei A., & Darrell, Trevor. 2017. Curiosity-driven Exploration by Self-supervised Prediction. *CoRR*, **abs/1705.05363**.

Pathak, Deepak, Gandhi, Dhiraj, & Gupta, Abhinav. 2019. Self-Supervised Exploration via Disagreement. *CoRR*, **abs/1906.04161**.

Pitis, Silviu, Chan, Harris, Zhao, Stephen, Stadie, Bradly C., & Ba, Jimmy. 2020. Maximum Entropy Gain Exploration for Long Horizon Multi-goal Reinforcement Learning. *CoRR*, **abs/2007.02832**.

Pong, Vitchyr H., Dalal, Murtaza, Lin, Steven, Nair, Ashvin, Bahl, Shikhar, & Levine, Sergey. 2019. Skew-Fit: State-Covering Self-Supervised Reinforcement Learning. *CoRR*, **abs/1903.03698**.

Rajeswar, Sai, Mazzaglia, Pietro, Verbelen, Tim, Piché, Alexandre, Dhoedt, Bart, Courville, Aaron, & Lacoste, Alexandre. 2023. *Mastering the Unsupervised Reinforcement Learning Benchmark from Pixels*.

Salge, Christoph, Glackin, Cornelius, & Polani, Daniel. 2013. Empowerment - an Introduction. *CoRR*, **abs/1310.1863**.

Sekar, Ramanan, Rybkin, Oleh, Daniilidis, Kostas, Abbeel, Pieter, Hafner, Danijar, & Pathak, Deepak. 2020. Planning to Explore via Self-Supervised World Models. *CoRR*, **abs/2005.05960**.

Shyam, Pranav, Jaskowski, Wojciech, & Gomez, Faustino. 2018. Model-Based Active Exploration. *CoRR*, **abs/1810.12162**.

Strouse, DJ, Baumli, Kate, Warde-Farley, David, Mnih, Vlad, & Hansen, Steven. 2021. Learning more skills through optimistic exploration. *CoRR*, **abs/2107.14226**.

Yarats, Denis, Fergus, Rob, Lazaric, Alessandro, & Pinto, Lerrel. 2021. Reinforcement Learning with Prototypical Representations. *CoRR*, **abs/2102.11271**.

A Visualization of Bandit Objectives

Figure 4 details the policy and reward used in each bandit problem.

B Visualization of Skillset Parameter Policy Actor-Critic

Figure 5 illustrates how the policy f_λ connects to the parameter-specific critics Q_{κ_i}.

C Visualizations

Figure 6 shows that our approach effectively optimizes empowerment in the stochastic four rooms pick-and-place domain by visualizing the entropies and conditional entropies of the learned skillset.

Figure 7 visualizes the entropy terms for the RGB QR code navigation task. Note that in this implementation, we used a four dimensional skill space which is twice as large as the two dimensional underlying state space. Per the image on the right, the algorithm corresponds by only using two dimensions of the latent state space, which is why you see the horizontal lines formed by the variational posterior.

Figure 8 visualizes the entropy terms for the RGB QR code pick-and-place task. Per the images the agent learns abstract skills to move itself and the object to a specific regions of the underlying (x,y) space.

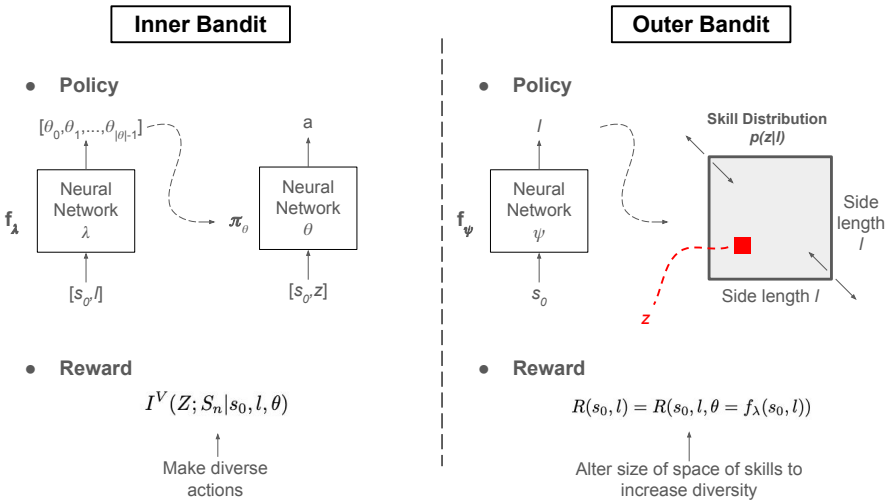

Figure 4: (Left) Inner bandit f_λ learns to output more diverse θ given a start state s_0 and skill distribution parameter l. θ is used as the weights and biases for the skill-conditioned policy neural network. (Right) Outer bandit f_ψ learns to output l, which is the side length of the d-dimensional cube that represents the uniform distribution over skills (in this case, skills are sampled from a $d = 2$ square). If the agent can execute a greater diversity of skills with a larger l, f_ψ will be encouraged to increase l. But f_ψ can also maintain l or shrink it if that produces a more diverse skillset with larger $R(s_0, l)$.

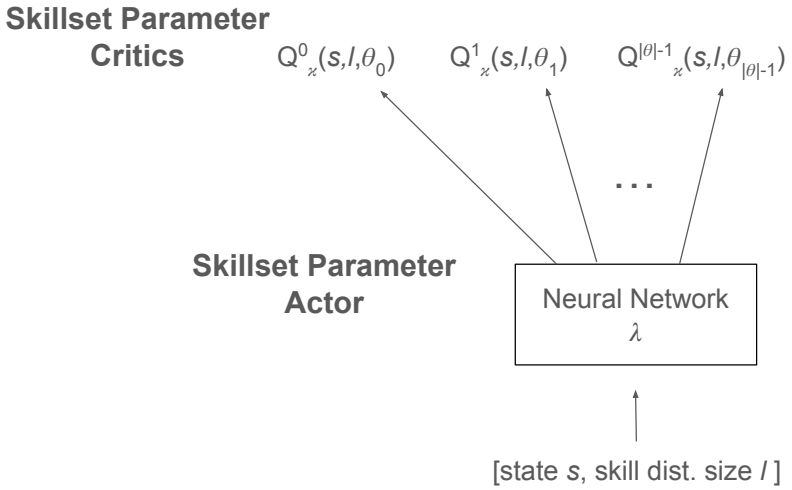

Figure 5: Visualization of the parameter-specific actor-critic structure used to update f_λ that outputs θ. Each parameter-specific critic Q_{κ_i} approximates how the reward $I^V(Z; S_n|s_0, l, \theta)$ reacts to small changes in the i-th parameter of θ. Together the parameter-specific critics can mimic having a regular critic $Q_\kappa(s_0, l, \theta)$.

Figure 9 shows the state coverage of the GCRL agent in the stochastic four rooms task. Per the image, the agent only learns skills to target the corners of rooms.

Figure 10 shows the behavior of the skills learned by the GCRL agent. Each skill simply moves in the direction of the room of the goal state regardless of where in the room the goal is. For instance, given the purple goal (shown by purple square) in the lower left of the top right room, the agent just moves to the top right, which you can see by the purple circles in the top right room and bottom left room (hard to see). Similarly for the dark blue goal, the agent just moves to the top left, which you can see by the blue circles in the top left of the various rooms.

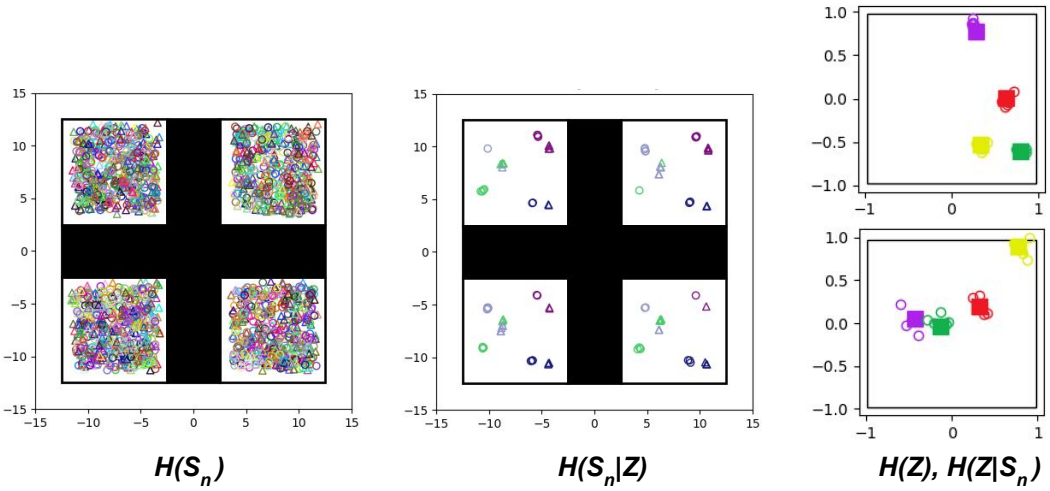

Figure 6: Images show the entropy visualizations for the stochastic four rooms pick-and-place domain. The left image shows the skill-terminating states s_n that result from 1000 skills uniformly sampled from the learned skill space. The near uniform coverage of the state space shows that $H(S_n)$ is large. The middle image focuses on four skills, uniformly sampled from the skill space, and for each skill shows 12 samples of skill-terminating states. Per the image, each skill targets an abstract state representing an offset from the center of a room for both the agent and object, showing that $H(S_n|Z)$ is low. The right image focuses on four skills and shows 5 samples from the variational posterior $q_\psi(z|s_0, l, \theta, s_n)$. Per the image, the samples form a narrow distribution around the executed skill, showing that $H(Z|S)$ is low.

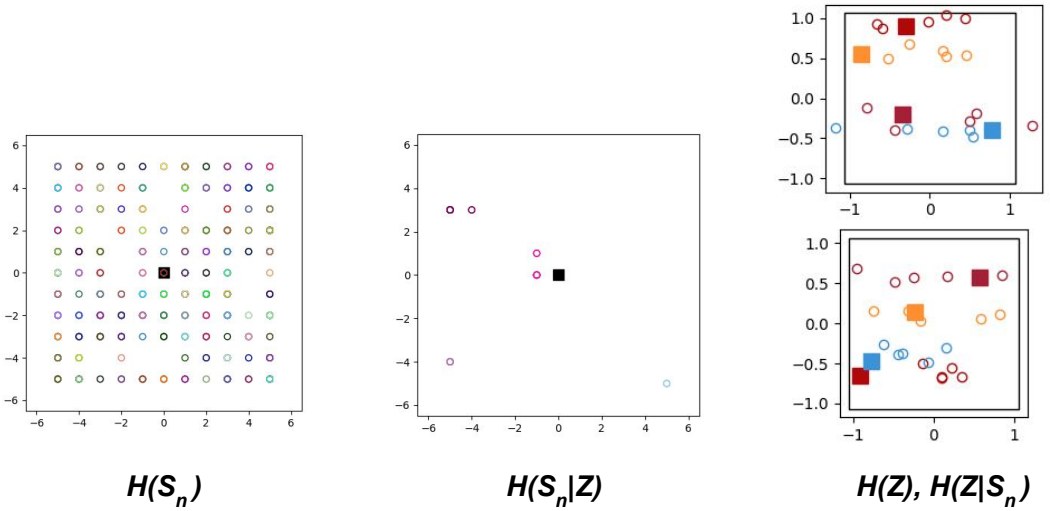

Figure 7: Entropy visualizations for the RGB QR code navigation task. Left image visualizes $H(S_n)$ by marking the skill-terminating states s_n produced by executing 1000 samples of skills from the learned skill space. Center image visualizes $H(S_n|Z)$ by executing four skills 12 times each and recording the skill-terminating states. Each skill targets an abstract (x,y) position. The right image shows samples from the variational posterior distribution. Note that in this case, the latent space is four dimensional even though the underlying state space is two dimensional. Because the agent does not need those extra dimensions, you see the horizontal lines in the variational posterior visualization.

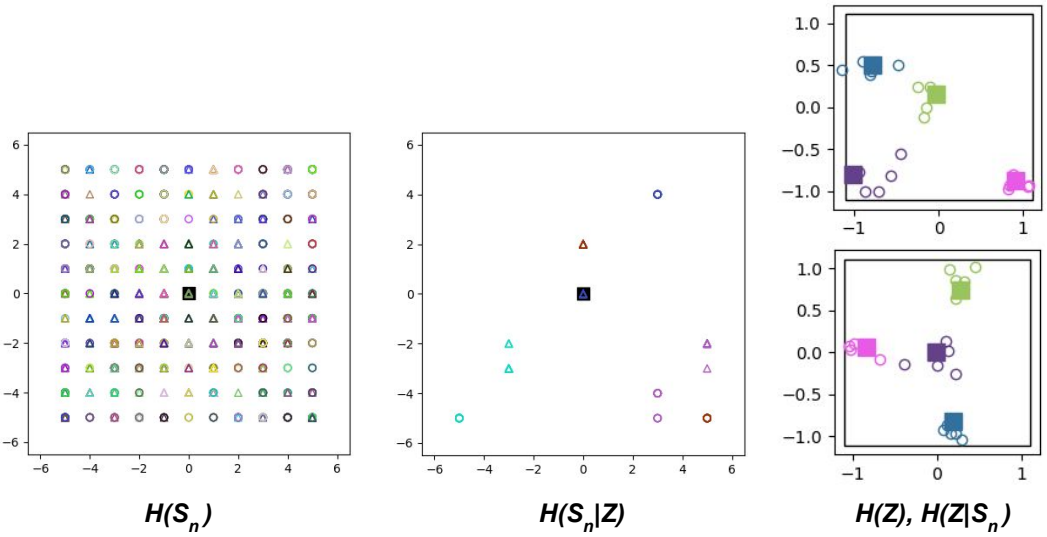

Figure 8: Entropy visualizations for the RGB QR code pick-and-place tasks. Left image visualizes $H(S_n)$ by marking the skill-terminating states s_n produced by executing 1000 samples of skills from the learned skill space. Center image visualizes $H(S_n|Z)$ by executing four skills 12 times each and recording the skill-terminating states. Each skill targets an abstract (x,y) position. The right image shows samples from the variational posterior distribution.

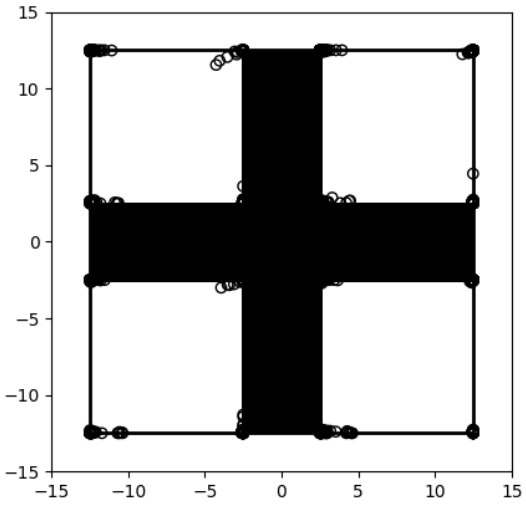

Figure 9: GCRL state coverage in stochastic four rooms domain.

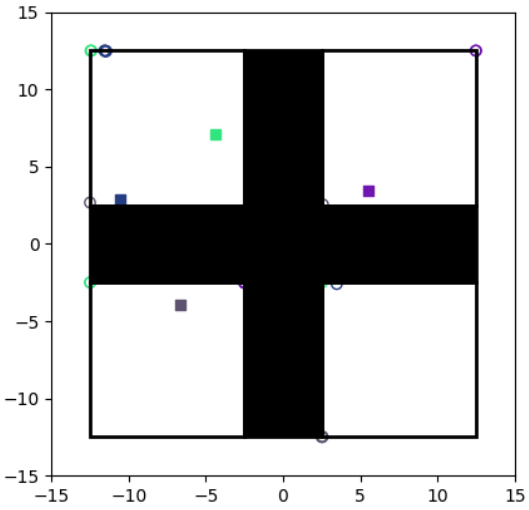

Figure 10: Image shows the skill-terminating states (empty circles) from four randomly selected goal states (filled squares). Each skill just moves in the direction of the room the skill is in.

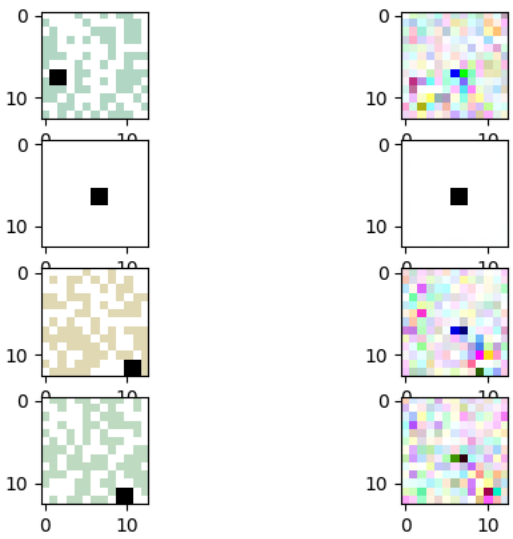

Figure 11: Image shows a sample of the VAE results in the RGB QR code navigation task. The left column shows sample images from the environment and the right column shows the results when those samples are encoded and then decoded. The VAE was able to decode the initial state of the environment, which is just a white background with the agent in the center, but struggled for other states.

Figure 11 shows a sample of the VAE results in the RGB QR navigation domain. The VAE was able to decode the start state of a white background with the black agent in the center, but struggled with the other states. The area in which the agent was located would sometimes have slightly darker pixels but not enough to distinguish the state.

D Environment Details

We implemented the following environments.

1. **Stochastic Four Rooms Navigation:** This domain consists of a two-dimensional point agent in an environment with four walled rooms. The domain is highly stochastic because after each action, the agent is placed in a room uniformly sampled. The location of the agent after executing an action is the sum of (i) the agent's $(\Delta x, \Delta y)$ position relative to the center of its current room, (ii) the next action, (iii) small gaussian noise, and (iv) the center of the new room. Thus, the "abstract" state in this domain is the $(\Delta x, \Delta y)$ position relative to the center of the current room. The observation space is two-dimensional consisting of the (x, y) position of the agent. The action space is also two-dimensional and consists of the the $(\Delta x, \Delta y)$ action. The action space for each dimension is the range $[-1, 1]$.
2. **Stochastic Four Room Pick-and-Place:** This is the same environment as the navigation task except there is now an object that can be moved. The observation space is four-dimensional consisting of the two-dimensional positions of the agent and object. The action space is also four-dimensional. The first two dimensions determine the change in the relative position of the agent. The final two dimensions determine the change in the relative position of the object. However, this action is only applied if both the x and y positions of the object are within 2.5 units of the agent. The agent starts in the same position as the object.
3. **RGB QR Code Navigation:** In this domain an agent learns to navigate amid a continually changing RGB-colored QR code background. The underlying dynamics in this environment are simple. The environment state, not visible to the agent, is the two-dimensional position of the agent. Actions are discrete and consist of a horizontal movement (i.e., move west, move east, or no change) and a vertical movement (i.e., move north, move south, or no change). In the underlying state space, the transition dynamics are deterministic. However, the observation space and dynamics are more complex. The observation space is high-dimensional consisting of $3x13x13$ RGB images (i.e., 507 total pixels), in which the dimensions represent (number of channels, image length, image width). The agent in every image is represented by a 3x2x2 black pixel. The domain is highly stochastic as the RGB-colored QR code background completely changes after each action.
4. **RGB QR Code Pick-and-Place:** This environment is the same as the navigation task except there is a now an object that can be moved. The object is always represented by a yellow square of pixels. In additional to the agent position changes, agent actions now also include a horizontal object position change (i.e., no change, move west, move east) and a vertical object position change (i.e., no change, move north, move south). The box can be moved if the both the underlying x and y positions of the box are within two units of the the agent. The agent starts in the same position as the object.

E Baseline Details

For the GCRL comparison, in the low-dimensional stochastic four rooms domains, we compared against the variant of GCRL that is a lower bound to Empowerment (see section 2.3 for details on the specific reward function this variant uses). The goal distribution is set to the distribution of all reachable state (e.g., all possible agent (x, y) positions in the stochastic four rooms navigation task). For the higher dimensional QR code tasks, we implemented Reinforcement learning with Imagined Goals (RIG) Nair *et al.* (2018). RIG is an unsupervised GCRL method that combines representation learning and GCRL. RIG uses a VAE to separately learn an encoder that maps state to distributions over skills and a decoder that maps latent states to distributions over observations. RIG then performs GCRL in the learned embedding space (i.e, the agent learns skills that target specific latent states). Because the focus of this paper is not exploration, we make it easier on the representation learning component of RIG and provide it with a large dataset of reachable observations (e.g., images of the

Table 1: Table shows various measures of the parallel computation demands for each environment.

TASK	$\|\theta\|$	UPDATE TIME (S)	GPU NOTES
FOUR ROOMS NAV	386	37	1 A100 40GB OR 2 H100 80GB SXM5
FOUR ROOMS PICK	484	56	1 A100 40GB OR 2 H100 80GB SXM5
RGB QR NAV	2528	10	8 A100 80GB SXM
RGB QR PICK	2528	10	8 A100 80GB SXM

agent and object in a large variety of positions in the pick-and-place QR code environment.) The goal distribution for the GCRL phase is the prior distribution $p(z)$ from the VAE component.

F Parallel Computation Demands

Table 1 provides some data on the parallel computation demands of our approach for each task. $|\theta|$ is the number of parameters in the skill-conditioned policy (equivalent, the dimensionality of the output space of the skillset parameter policy). Update Time reflects the time (in seconds) required to complete one whole update step (i.e., one iteration of the Repeat loop in Algorithm **??**). Note that the update times shown for the four rooms tasks were when using a single A100 40GB device, while the update times for the RGB QR tasks reflect 8 A100 80GB SXM GPUs. When we used multiple GPUs, the update times were roughly $1/\text{Num GPUs}$ of the original time with a single GPU.